ACCURACY MODEL FOR RECOGNITION SIGNAL PROCESSING ENGINES

FIELD OF THE INVENTION

[01] The present invention relates to computer learning software used for, e.g., recognition of handwriting, speech and other forms of human input. In particular, the present invention relates to evaluating the accuracy of signal processing by such software.

[02] BACKGROUND OF THE INVENTION

[03] Computers accept human input in various ways. One of the most common input devices is the keyboard. Additional types of input mechanisms include mice and other pointing devices. Although useful for many purposes, keyboards and mice (as well as other pointing devices) sometimes lack flexibility. For these and other reasons, various alternative forms of input have been (and continue to be) developed. For example, electronic tablets or other types of electronic writing devices permit a user to provide input in a manner very similar to conventional writing. These devices typically include a stylus with which a user can write upon a display screen. A digitizer nested within the display converts movement of the stylus across the display into an "electronic ink" representation of the user's writing. The electronic ink is stored as coordinate values for a collection of points along the line(s) drawn by the user. Speech is another alternative input form. Typically, the user speaks into a microphone, and the user's speech is digitized and recorded.

[04] Before the information conveyed by speech or ink can be usefully manipulated by a computer, the speech or ink must usually undergo recognition processing. In other words, the graphical forms (ink) or sounds (speech) created by the user are analyzed by various algorithms to determine what characters (e.g., letters, numbers, symbols, etc.) or words the user intended to convey. Typically, the ink or speech is converted to Unicode, ASCII or other code values for what the user has recorded.

[05] Various systems have been developed to recognize handwriting and speech input. In many cases, recognition algorithms involve isolating individual features of ink or speech input. These features are then compared to previously-generated prototype features. In particular, numerous samples of ink or speech are initially collected and used to create a database of prototype features against which an unknown ink or speech sample is compared. Based on the degree of similarity (or dissimilarity) between an input sample and one or more prototype features, one or more recognition results are generated for a particular user input.

[06] Accuracy, or the closeness of a recognition result to what the user intended to convey, is an important criterion by which recognition systems are evaluated. For obvious reasons, a recognition engine must be reasonably accurate in order to be useful. However, "accuracy" is not easily defined or quantified. For example, a recognizer may work quite well in circumstance A, but not work well in circumstance B. Depending on what circumstances A and B are, this may or may not be significant. If a recognizer works well in circumstances that are very important to the end user, and only works poorly in circumstances that are of little consequence, the overall accuracy of the recognizer might be considered good. Conversely, a recognizer that provides highly accurate results in obscure circumstances but does not work well in more important circumstances might be considered inaccurate overall.

[07] Accordingly, there remains a need for systems and methods of modeling the accuracy of signal processing engines used for, e.g., recognition of speech, handwriting and other complex forms of input.

[08] SUMMARY OF THE INVENTION

[09] The present invention addresses the above and other challenges associated with modeling the accuracy of a computer learning signal processing engine used for, e.g., handwriting or speech recognition. Signals to be processed are categorized based on signal characteristics such as physical aspects, context, conditions under which the

signals were generated and source, and/or based on other variables. Categorized sets of signals are processed, and an accuracy for each set calculated. Weights are then applied to accuracy values for the sets, and the weighted values summed. In some cases, certain sums are then weighted and further summed.

[10] In one illustrative embodiment, the invention includes a method for evaluating a computer learning signal processing engine. The method includes selecting a plurality of variables having values characterizing multiple signals to be processed. A first group of signal sets is identified, each signal set of the first group having an associated range of values for a variable of the plurality corresponding to the first group. An accuracy score for each signal set of the first group is calculated using the signal processing engine to be evaluated. Weight factors are applied to the accuracy scores for the first group signal sets. Each weight factor represents a relative importance of one of the associated ranges of values for the first variable. The weighted accuracy scores for first group signal sets are then summed to yield a first summed accuracy score. The method further includes identifying additional groups of signal sets, each group having a corresponding variable of the plurality of variables, each signal set of a group having an associated range of values for the corresponding variable. Accuracy scores for each signal set of each additional group are also calculated using the signal processing engine to be evaluated. Weight factors are applied to the accuracy scores for the signal sets of the additional groups. The weight factors within each of the additional groups represent relative importance of associated ranges of values for the variable corresponding to the group. The weighted accuracy scores within each of the additional groups are summed to yield additional summed accuracy scores, and the summed accuracy scores are further combined.

[11] These and other features and advantages of the present invention will be readily apparent and fully understood from the following detailed description of various embodiments, taken in connection with the appended drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[12] FIG. 1 is a block diagram illustrating operation of at least some embodiments of the invention.

[13] FIG. 2 is a table showing data used in at least one embodiment of the invention.

[14] FIG. 3 is a diagram showing an accuracy model according to at least one embodiment of the invention.

[15] FIGS. 4-15 are diagrams showing additional details of nodes of the accuracy model of FIG. 3.

[16] FIG. 16 is a diagram showing application of a transform function to a node of the accuracy model of FIG. 3.

[17] FIGS. 17 and 18 are diagrams showing addition of nodes to an accuracy model according to various embodiments of the invention.

[18] FIG. 19 is a table showing data used in at least one other embodiment of the invention.

[19] FIG. 20 is an accuracy model according to at least one other embodiment of the invention.

[20] FIG. 21 is a block diagram of a general-purpose digital computing environment that can be used to implement various aspects of the invention.

[21] DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[22] Embodiments of the invention provide a deterministic model for computing an overall accuracy value for computer learning signal processing systems, such as are used for

handwriting or speech recognition. These systems are analyzed using numerous samples of user input. Various aspects of the samples and of the circumstances of the sample generation are identified. Accuracy values of the recognizer are then determined with regard to sets of the samples grouped by these identified aspects. These accuracy values are then weighted and combined to obtain an overall accuracy value for the recognizer (or other computer learning signal processing system). Although the invention is described using handwriting recognition and speech recognition as examples, this is only for purposes of illustrating operation of the invention. The invention is not limited to implementations related to handwriting and speech recognition.

[23] Aspects of the invention may be implemented with program modules or other instructions that can be executed on a computing device. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Because the invention may be implemented using software, an example of a general purpose computing environment is included at the end of the detailed description of the preferred embodiments. Embodiments of the invention are in some instances described using examples based on user interfaces and software components found in the MICROSOFT WINDOWS XP Tablet PC Edition operating system ("XP Tablet") available from Microsoft Corporation of Redmond, Washington, as well as by reference to application programs used in conjunction with XP Tablet. However, any operating system or application program named is only provided so as to provide a convenient frame of reference for persons skilled in the art. The invention is not limited to implementations involving a particular operating system or application program.

[24] FIG. 1 is a block diagram showing operation of at least some embodiments of the invention. In order to evaluate the accuracy of a recognition signal processing engine, sample input is needed. Although actual sample collection is not part of all

embodiments of the invention, such collection is shown in FIG. 1 for purposes of explanation. As shown in block 10, sample inputs are collected for processing by a recognition engine to be evaluated. In the case of handwriting recognition, samples are obtained from a diverse group of users providing various types of input under various conditions. For example, handwriting samples are gathered from users having different demographic characteristics (e.g., male, female, age group, native language, etc.). In some cases, multiple users may be asked to write the same text. In other cases, different users in the same group may be asked to write different text. For example, one group of users may be asked for sample e-mail addresses, another group asked for sample text that might be input to a word processing program, etc. In some embodiments, the sample collection shown in block 10 is performed specifically for use in evaluating a particular recognition engine. In other embodiments, block 10 generically refers to input samples that have been previously collected and are used for evaluation of multiple recognition engines, or that have been obtained for other purposes and adapted to evaluation of a recognition engine. Notably, the invention is not restricted to implementations in which a single set of data is used to obtain an overall accuracy score. For example, data collection from various sources could be combined. Older pre-existing data could be augmented with additional data, collections of data gathered for various unrelated reasons could combined, etc. In at least one embodiment, data is collected using tools and environments other than those which will be used to implement a recognizer being evaluated.

[25] In block 12, collected input samples are classified according to numerous variables. Although shown as a separate block in FIG. 1 for convenience, the activities of blocks 10 and 12 could occur simultaneously. For example, as input samples are obtained, descriptive information about the sample could also be collected. FIG. 2 further illustrates collection and classification of sample inputs, and more specifically, sample handwriting input. Each row of table 20 corresponds to a different user input sample. Each column corresponds to a different variable used for accuracy analysis. Each cell of table 20 thereby contains a value for each sample for a particular variable. Actual

values are not included in table 20; instead, the vertical dots indicate that numerous entries are contained in each column. Table 20 of FIG. 2 is for purposes of illustration, and an actual table is not necessarily generated in operation of the invention. However, data of the type shown in table 20, which data may be organized within a database or other data management program, is accessed in connection with at least some embodiments of the invention.

[26] The first column of table 20 ("Sample ID") is merely an identifier of a sample, and may be a sequential record number assigned automatically as samples are collected and/or stored. The second column ("Information") corresponds to the actual information the user intends to convey with his or her handwriting. If the user is attempting to write an e-mail address, for example, the value for "Information" could be something in the form "emailrecipient@domainname.com".

[27] The next column of table 20 corresponds to demographic data regarding the provider of the sample. For example, samples may be classified by sample provider gender, by age or age group (e.g., 10-20, 20-35, etc.), by native language, by whether the user is left- or right-handed, etc.

[28] The next column of table 20 corresponds to the input scope of the sample. In many cases, the input scope is analogous to the context of the information the user is attempting to convey by writing. Possible values for input scope include full name, given name, middle name, surname, nickname, e-mail address, computer system username, an Internet Uniform Resource Locater (URL), a postal address, a postal code, a telephone number, etc. These are only examples, however, and a large number of other possibilities exist. For example, a particular recognition engine may be designed for use by medical personnel. In such a case, input scope values could include such things as patient name, drug name, drug dosage, etc.

[29] The next column of table 20 corresponds to the spacing between components of the input sample. Some handwriting recognition engines use space between ink strokes

(or collections of ink strokes) in connection with determining whether the user has written a separate letter, separate words, etc. Different individuals tend to write with different amounts of space between letters and/or words. Values for space could be in absolute terms (e.g., the number of pixels between separate stroke groupings) or could be relative (e.g., ratio of spacing between stroke groups to total length of ink). In some embodiments, spacing also has a temporal component, e.g., the delay between writing portions of an ink sample.

[30] The next column corresponds to scaling of the input. In some embodiments, this variable relates to the size of the input. For example, a value for this variable may be the length, the height and/or the length/height ratio of an input sample. As another example, this variable may correspond to the amount by which an input sample is automatically enlarged or reduced as part of recognizer processing. More specifically, many recognition engines expand an electronic ink sample to fill a standard size area as part of feature extraction and comparison. In such a case, a scaling value may be the amount by which the original sample must be enlarged to fit within the standard area.

[31] The next column corresponds to the user scenario in which the input is generated. In some embodiments, the user scenario includes the specific software application(s) used and/or the operations performed by the user in connection with creating the input. Examples of different user scenarios include:

> User opens the Tablet PC Input Panel (TIP) of XP Tablet, user then opens a file in a text application to receive text input from the TIP, and user then creates an electronic ink word in the TIP and saves the ink in the opened file.
>
> User opens a TIP, user then opens a file in a word processing application to receive the text results from the TIP, user then creates electronic ink in the TIP and sends same to the word processing application, user then instantiates and uses a correction user interface in the word processing application.

User opens a note-taking utility application (such as WINDOWS Journal utility of XP Tablet), user then creates an electronic ink note title, user then saves a file having the title as a file name.

User opens a note-taking utility application, user then creates an electronic ink paragraph, user then selects the paragraph, user then saves the ink paragraph as text.

The foregoing are merely examples, and numerous other user scenarios can be defined.

[32] The next column corresponds to content, and represents various additional types of data regarding input samples. In some embodiments, values for content include identification of one or more software programs or program components from which the sample was obtained. In some embodiments, "content" may include the identity of a program that malfunctions during ink collection. In other embodiments, content values are associated with characteristics of a language model used by a recognizer. In still other embodiments, content values relate to aspects of the input suggesting that a user has used extra care to make the input neat and recognizable (e.g., writing in all capital letters).

[33] The remaining columns of table 20 are labeled "format" and "angle." "Format" corresponds to the file format(s) in which the graphical and/or text components of an ink input is saved. Examples of file formats in which the graphical component of ink may be saved include bitmap files (.bmp), graphical interchange format files (.gif) and ink serialized format files (.isf). Examples of file formats in which text and/or metadata regarding ink may be saved include binary files (.bin), rich text format files (.rtf), hypertext markup language files (HTML) and extensible mark-up language files (XML). The angle column contains value(s) representing the angles between words, characters, or other ink strokes of an input sample.

[34] The variables shown in table 20 are merely examples of types of information associated with a user input. In some embodiments, one or more of the variables in table 20 are not tracked or used for computation of an overall accuracy score. In other embodiments, different variables are recorded and used for computing an overall accuracy score. Various combinations and sub-combinations of one or more variables are also within the scope of the invention. For example, user input could be sorted based on multiple demographic factors: male users aged 10-20, female users who are native Japanese speakers, left-handed male users aged 20-35 who are native English speakers, etc.

[35] Returning to FIG. 1, block 14 represents calculating the accuracy of a recognition engine as to sets of individual samples that are grouped by a particular variable. In at least one embodiment, the sets are grouped by values for the variables listed in table 20. For example, all of the input samples could be sorted by gender and handedness of the user, resulting in the following sets: right-handed females, left-handed females, right-handed males and left-handed males. A collective accuracy is then calculated for each of the four sets. Various measures of collective accuracy can be used. In one embodiment, collective accuracy for a set is computed according to Equations 1 through 3.

(Equation 1) $$\text{Collective Accuracy} = \sum_{i=1}^{m} \eta_i * \left(A_i + \frac{1}{1+D_i} \right)$$ where

(Equation 2) $$\sum_{i=1}^{m} \eta_i = \frac{1}{2m}$$ and where

(Equation 3) $$A_i = \begin{cases} 1 & \text{if word is top recognizer result} \\ 0 & \text{otherwise} \end{cases}$$

[36] The counter m in Equations 1 and 2 corresponds to the number of raw input words in a set provided to a recognizer for processing. Returning to the example of

gender/handedness groupings, if collective accuracy is computed for each set, m for one set equals the total number of raw input words generated by right-handed females; some of the samples may be entire paragraphs, some may be a few words (e.g., a full name, an address, etc.) and some may only be a single word (e.g., an e-mail address). The term D_i in Equation 1 is a distance function between the intended words (or characters or other components) in a set of user input samples and the results returned by the recognizer for those input samples. Often, a recognizer will return multiple results for an input, and D_i for a set is computed as a real value between zero and ∞ (infinity). In at least one embodiment, D_i is the sum, for an entire set, of the number of differences between an intended word (or word group) and all rearrangements of the results returned by the recognizer. In at least another embodiment, D_i is a ratio of the number of recognition results for the words (or other components) of a set divided by the number of input words (or other components) in a set. In still other embodiments, D_i is simply the ratio of the number of correctly recognized words (or characters or other components) in a set divided by the total number of words (or characters or other components) in a set. Various other measures of D_i can be used.

[37] The term η_i (Equation 2) is a factor allowing introduction of weights for different words in multi-word groups, thereby placing more emphasis on words for which correct recognition is deemed to be more important. For example, a double η_i value could be assigned to words that are the 1000 most-used words, are the 1000 most-suggested recognition results, etc.

[38] As indicated by Equation 3, A_i has a value of 1 if the correct recognition result for an input word is the first (or only) word suggested by a recognizer in a list of possible recognition results. Otherwise, A_i has a value of 0. In other embodiments, A_i is calculated in a different manner. For example, A_i in some embodiments equals 1 if the intended word is in the top N recognition results (where N is an integer), and is otherwise 0. In other embodiments, A_i can have values between 0 and 1.

[39] Block 16 of FIG. 1 corresponds to calculation of an overall (or total) accuracy A_T for a recognition (or other signal processing) engine. Overall recognizer accuracy is calculated using an accuracy model 30 such as is shown in FIG. 3. As explained below, inputs to accuracy model 30 include collective accuracy scores for sets (computed as described above), as well as weighting factors assigned to values for variables such as those listed in table 20 (FIG. 2). Accuracy model 30 of FIG. 3 is only one embodiment of an accuracy model according to the invention. As further described herein, many variations on the model of FIG. 3 are within the scope of the invention.

[40] Beginning at the right side of FIG. 3, summation node 32 corresponds to the overall accuracy (A_T) score. A_T is a weighted sum of two sub-components. In particular, A_T is a sum of a word accuracy A_W (computed in summation node 34) and user scenario accuracy A_{US} (computed in summation node 36). Word accuracy A_W is weighted by a factor γ_1 and user scenario accuracy A_{US} is weighted by a factor γ_2. Word accuracy A_W, computed at node 34, is the weighted sum of three sub-components: combined input scope accuracy A_{CIS} (computed at summation node 38), angle accuracy A_α (computed at summation node 40) and spacing accuracy A_{SP} (computed at summation node 42). Combined input scope accuracy A_{CIS} is weighted by the factor φ_1, angle accuracy A_α is weighted by the factor φ_2 and spacing accuracy A_{SP} is weighted by the factor φ_3.

[41] Combined input scope accuracy A_{CIS} is the weighted sum of accuracies for individual input scopes 1 through n computed at summation nodes 38_1 through 38_n. As used throughout this specification, "n" is a variable indicating an arbitrary number and does not necessarily represent the same value in all parts of model 30. In other words, n as used in one part of model 30 could represent a different arbitrary number than n as used in another part of model 30. Each individual input scope 1 through n is respectively weighted by factors ξ_1 through ξ_n. As shown by the vertical ellipsis between input scopes 2 through n, there may be many individual input scopes. Each

input scope is a weighted sum of five accuracy sub-components: demographics (computed at summation node 44), scaling (computed at summation node 46), angle (computed at summation node 48), content (computed at summation node 50) and format (computed at summation node 52). As explained in more detail below, each of the five accuracy subcomponents of an input scope corresponds to an accuracy score for a particular input scope. Each input scope 2 through n would have the same five accuracy subcomponents, but for the input scopes corresponding to nodes 38_2 through 38_n. Each of the demographic, scaling, angle, content and format accuracy sub-components of an input scope is weighted by respective factors π_1 through π_5.

[42] FIG. 4 further illustrates calculation of demographic accuracy (for input scope 1) at node 44. All input samples in a sample database (such as table 20 of FIG. 2) corresponding to input scope 1 are identified. For example, if input scope 1 is a postal address, all of the samples in which the user intended to write a postal address are identified. The identified samples are sorted into sets based on values for the demographic variable. Returning to a previous example, all input samples in which the user intended to write a postal address are in one embodiment sorted into four sets: right-handed females, left-handed females, right-handed males and left-handed males. Using, e.g., the formulae of Equations 1 through 3, a collective accuracy is then calculated for each set. The calculated collective accuracy for each set is then provided as an input (D1, D2, ... Dn) to the calculation of D(IS1), the demographic accuracy D for input scope 1.

[43] Each of these inputs is weighted by a respective weighting factor μ_1 through μ_n. Weighting factors μ_1 through μ_n, which may be determined in a variety of manners, total 1.0. In at least one embodiment of the invention, weights applied to inputs to a node of an accuracy model are based on the relative importance of the inputs to that node. In many cases, the weights are a function of the utility and/or the relevance (as perceived by a user) of a particular input. Weights may be determined based on user research, on usability studies, on product planning, on technical factors of a

recognizer or its target software or hardware environment, or on numerous other factors (or combinations of factors). Additional examples are provided below.

[44] In the case of the demographic variable in the example of FIGS. 3 and 4, each weighting factor represents the relative importance assigned to a particular demographic value. For example, a particular recognition engine may be targeted for implementation in a software product for which it is estimated that 75% of the customers will be female. Using values for overall percentages of the general population who are right handed (% right) versus left-handed (% left), weighting factors μ for node 44 could be computed as follows: μ_1 (right-handed females) = .75 * (% right); μ_2 (left-handed females) = .75 * (% left); μ_3 (right-handed males) = .25 * (% right); and μ_4 (left-handed males) = .25 * (% left). In other embodiments, weighting factors may be based on criteria other than a target market for a product and/or calculated in a different manner. After multiplying by a weighting factor μ, the collective accuracies D1-Dn for the sets are summed according to Equation 4.

(Equation 4) $$\text{Demographic accuracy (input scope 1)} = \text{D(IS1)} = \sum_{i=1}^{n} \mu_i D_i$$

The counter n in Equation 4 corresponds to the number of demographic value sets into which samples having input scope 1 are sorted. In the previous example of right- and left-handed males and females, n = 4.

[45] FIG. 5 illustrates calculation of scaling accuracy for input scope 1 at node 46. Notably, samples in the database of table 20 have multiple different characteristics; the recognition accuracy for a given sample may thus be included in the inputs to more than one node of model 30 (FIG. 3). All identified input samples corresponding to input scope 1, which inputs were previously sorted based on demographic values, are sorted into sets based on values (or ranges of values) for the scaling variable. For example, one set may correspond to samples scaled by factors between 1.0 and 1.1, another set to samples scaled by factors of 1.1 to 1.3, etc. Collective accuracy is then

calculated for each set. The calculated collective accuracy for each set is then provided as an input (S1, S2, ... Sn) to the calculation of S(IS1), the scaling accuracy S for input scope 1. Each of these inputs is then weighted by a respective weighting factor μ_1 through μ_n. Weighting factors μ_1 through μ_n, which may be different from the weighting factors μ_1 through μ_n used for other nodes, may also be determined in a variety of manners, and will also total 1.0. In the present example, each weighting factor represents the relative importance assigned to a particular value for scaling of an input sample. For example, a particular recognition engine may be targeted for implementation in a software product for which research shows most users write with relatively small letters, resulting in larger scaling by the recognizer. In such a case, higher values of scaling could be assigned a larger weight. Again, weighting factors may be based on criteria other than a target market for a product. As but one example, specific digitizer or other hardware with which a recognizer is used may suffer loss of resolution at higher scaling values, requiring assignment of a certain weight to higher scaling values. After multiplying by a weighting factor μ, the collective accuracies S1-Sn for the sets are summed according to Equation 5.

(Equation 5) $$\text{Scaling accuracy (input scope 1)} = S(IS1) = \sum_{i=1}^{n} \mu_i S_i$$

The counter n in Equation 5 corresponds to the number of scaling value sets into which samples having input scope 1 are sorted.

[46] FIG. 6 illustrates calculation of angle accuracy for input scope 1 at node 48. All identified input samples corresponding to input scope 1 are sorted into sets based on values (or ranges of values) for the angle variable. In one embodiment, the value for the angle variable is the angle between adjacent words relative to the x axis in an (x,y) coordinate system. Collective accuracy is then calculated for each set. The calculated collective accuracy for each set is then provided as an input (α1, α2, ... αn) to the calculation of α(IS1), the angle accuracy α for input scope 1. Each of these inputs is weighted by a respective weighting factor μ_1 through μ_n. Weighting factors μ_1

through μ_n, which may be different from the weighting factors μ_1 through μ_n used for other nodes, may also be determined in a variety of manners, and will also total 1.0. In the present example, each weighting factor represents the relative importance assigned to a particular value for angle(s) of an input sample. For example, data may show that a majority of users tend to write with a particular range of angles between words, and values within that range may be assigned a larger weight. As before, weighting factors may be based on criteria other than a target market for a product. After multiplying by a weighting factor μ, the collective accuracies $\alpha 1$- αn for the sets are summed according to Equation 6.

(Equation 6) $$\text{Angle accuracy (input scope 1)} = \alpha(\text{IS1}) = \sum_{i=1}^{n} \mu_i \alpha_i$$

The counter n in Equation 6 corresponds to the number of angle value sets into which samples having input scope 1 are sorted.

[47] FIG. 7 illustrates calculation of content accuracy for input scope 1 at node 50. All identified input samples corresponding to input scope 1 are sorted into sets based on values (or ranges of values) one or more content variables. Collective accuracy is then calculated for each set. The calculated collective accuracy for each set is then provided as an input (C1, C2, ... Cn) to the calculation of C(IS1), the content accuracy C for input scope 1. Each of these inputs is weighted by a respective weighting factor μ_1 through μ_n. Weighting factors μ_1 through μ_n, which may be different from the weighting factors μ_1 through μ_n used for other nodes, may also be determined in a variety of manners, and will also total 1.0. Each weighting factor represents the relative importance assigned to a particular value for a given content variable or variable combination. After multiplying by a weighting factor μ, the collective accuracies C1-Cn for the sets are summed according to Equation 7.

(Equation 7) $$\text{Content accuracy (input scope 1)} = \text{C(IS1)} = \sum_{i=1}^{n} \mu_i C_i$$

The counter n in Equation 7 corresponds to the number of content value sets into which samples having input scope 1 are sorted.

[48] FIG. 8 illustrates calculation of format accuracy for input scope 1 at node 52. All identified input samples corresponding to input scope 1 are sorted into sets based on value(s) of one or more format variables. For example, one set may include samples stored in graphic format Q and text format R, another set may include samples stored in graphic format Q and text format P, etc. Collective accuracy is then calculated for each set. The calculated collective accuracy for each set is then provided as an input (F1, F2, ... Fn) to the calculation of F(IS1), format accuracy F for input scope 1. Each of these inputs is weighted by a respective weighting factor μ_1 through μ_n. Weighting factors μ_1 through μ_n, which may be different from the weighting factors μ_1 through μ_n used for other nodes, may also be determined in a variety of manners, and will also total 1.0. In the present example, each weighting factor represents the relative importance assigned to a particular value for a format variable or variable combination. For example, one file format (or file format combination) may predominate for technical or marketing reasons, and thus be given a larger weight. After multiplying by a weighting factor μ, the collective accuracies F1-Fn for the sets are summed according to Equation 8.

(Equation 8) $$\text{Format accuracy (input scope 1)} = \text{F(IS1)} = \sum_{i=1}^{n} \mu_i F_i$$

The counter n in Equation 8 corresponds to the number of format value sets into which samples having input scope 1 are sorted.

[49] FIG. 9 illustrates calculation of accuracy for input scope 1 (A_{IS1}) at node 38_1. In particular, node 38_1 sums the weighted outputs of nodes 44 (demographic accuracy D(IS1)), 46 (scaling accuracy S(IS1)), 48 (angle accuracy α(IS1)), 50 (content accuracy C(IS1)) and 52 (format accuracy F(IS1)), as shown in Equation 9.

(Equation 9)

$$A_{IS1} = \Sigma [\pi_1 * D(IS1) + \pi_2 * S(IS1) + \pi_3 * \alpha(IS1) + \pi_4 * C(IS1) + \pi_5 * F(IS1) + ... + \pi_n * X(IS1)]$$

The accuracies D(IS1), S(IS1), α(IS1), C(IS1) and F(IS1) are weighted by respective weight factors π_1 through π_5. Each weight π corresponds to an importance assigned to a particular accuracy subcomponent of input scope 1. If, for example, accuracy for input scope 1 varies widely among different demographic sets, but varies little among sets based on angle between adjacent words, demographic accuracy D_{IS1} could be assigned a larger weight relative to angle accuracy α_{IS1}. Equation 9 includes additional terms beyond those supplied by nodes 44 through 52. In particular, Equation 9 continues with the expansion "+ ... + π_n*X(IS1)". As discussed in more detail below, other embodiments may have additional accuracy subcomponents for each input scope (shown generically as X(IS1)) weighted by a separate weighting factor (π_n). In the embodiment of FIG. 3, there are no other sub-components (i.e., π_6 through π_n equal 0). As with the weighting factors μ at each of nodes 44 through 52, the values of π_1 through π_n total 1.0.

[50] Accuracies for additional input scopes 2 (A_{IS2}, node 38_2) through n (A_{ISn}, node 38_n) are calculated in a similar manner. Specifically, samples corresponding to a particular input scope are identified, and accuracy sub-components D(IS), S(IS), α(IS), C(IS) and F(IS) calculated as described with regard to nodes 44 through 52. Notably, when calculating the accuracy subcomponents for a different input scope, the individual weighting factors at a particular node may change. In other words, μ_1 for D(IS1) may have a different value than μ_1 for D(IS2). The weighting factors μ_1 through μ_n for each node will still total 1.0, however. The accuracy sub-components D(IS), S(IS), α(IS), C(IS) and F(IS) are then multiplied by weighting factors π_1 through π_5. As with weighting factors for the same sub-component in different input scopes, the weighting factors π_1 through π_5 may also have different values for a different input scope. In

other words, π_1 for input scope 1 may have a different value than π_1 for input scope 2. The weighting factors π_1 through π_n for each input scope will total 1.0.

[51] FIG. 10 illustrates calculation of combined input scope accuracy A_{CIS} at node 38. Node 38 sums the weighted outputs of nodes 38_1 through 38_n, as shown in Equation 10.

(Equation 10) $$A_{CIS} = \sum_{i=1}^{n} \xi_i A_{ISi}$$

The counter n in Equation 10 corresponds to the number of input scope accuracies provided as inputs to node 38. Each input scope accuracy A_{ISi} is weighted by respective weight factors ξ_1 through ξ_n. Each weight ξ corresponds to an importance assigned to a particular input scope. For example, a recognizer targeted for use in an application program used mainly for e-mail could heavily weight e-mail addresses and other input scopes related to e-mail. The weighting factors ξ_1 through ξ_n for each input scope will total 1.0.

[52] FIG. 11 illustrates calculation of angle accuracy A_α at node 40. Node 40 calculates angle accuracy similar to node 48, but is not restricted to a specific input scope. All input samples are sorted into sets based on values (or ranges of values) for the angle variable. The sets may or may not be based on the same values (or value ranges) used for angle accuracy calculation for a specific input scope. Collective accuracy is then calculated for each set. The calculated collective accuracy for each set is then provided as an input ($\alpha 1$, $\alpha 2$, ... αn) to the angle accuracy calculation for all input scopes. Each of these inputs is weighted by a respective weighting factor τ_1 through τ_n. Weighting factors τ_1 through τ_n total 1.0. Each weighting factor τ represents the relative importance assigned to a particular value for an angle between words across different input scopes. Weighting factors τ_1 through τ_n may be determined in a variety of manners. For example, data may show that a majority of users tend to write with a particular range of angles between words in many different input scopes, and values

within that range may be assigned a larger weight. As before, weighting factors may be based on criteria other than a target market for a product. After multiplying by a weighting factor τ, the collective angle accuracy α for each of the sets is summed according to Equation 11.

(Equation 11) $$\text{Angle accuracy (all input scopes)} = A_\alpha = \sum_{i=1}^{n} \tau_i \alpha_i$$

The counter n in Equation 11 corresponds to the number of angle value sets into which samples are sorted.

[53] FIG. 12 illustrates calculation of spacing accuracy at node 42. Input samples across multiple input scopes are sorted into sets based on values (or ranges of values) for one or more spacing variables. Collective accuracy is then calculated for each set. The calculated collective accuracy for each set is then provided as an input (SP1, SP2, ... SPn) to the spacing accuracy calculation. Each of these inputs is weighted by a respective weighting factor σ_1 through σ_n. Weighting factors σ_1 through σ_n total 1.0. Each weighting factor σ represents the relative importance assigned to a particular value for spacing between words across different input scopes, and may be determined in a variety of manners. For example, data may show that a majority of users tend to write with a particular range of spaces between words, and values within that range may be assigned a larger weight. After multiplying by a weighting factor σ, the collective spacing accuracy SP for each of the sets is summed according to Equation 12.

(Equation 12) $$\text{Spacing accuracy (all input scopes)} = A_{SP} = \sum_{i=1}^{n} \sigma_i SP_i$$

The counter n in Equation 12 corresponds to the number of spacing value sets into which samples are sorted.

[54] FIG. 13 illustrates calculation of word accuracy A_W at node 34. In particular, node 34 sums the weighted outputs of nodes 38 (input scope accuracy A_{IS}), 40 (angle accuracy A_α) and 42 (spacing accuracy A_{SP}), as shown in Equation 13.

(Equation 13) $$A_W = \Sigma [\varphi_1 * A_{IS} + \varphi_2 * A_\alpha + \varphi_3 * A_{SP} + ... + \varphi_n * Z( )]$$

The accuracies A_{IS}, A_α, and A_{SP} are weighted by respective weight factors φ_1 through φ_3. Each weight φ corresponds to an importance assigned to a particular accuracy subcomponent of word accuracy A_W. If, for example, recognition accuracy varies widely based on spacing between words but varies little based on angle between words, spacing accuracy A_{SP} could be assigned a larger weight relative to angle accuracy A_α. Equation 13 includes additional terms beyond those supplied by nodes 38 through 42. In particular, Equation 13 continues with the expansion "+ ... + φ_n*Z( )". Other embodiments may have additional subcomponents for word accuracy A_W (shown generically as Z( )) weighted by a separate weighting factor (φ_n). In the embodiment of FIG. 3, there are no other sub-components of word accuracy A_W (i.e., φ_4 through φ_n equal 0). The values of φ_1 through φ_n total 1.0.

[55] FIG. 14 illustrates calculation of user scenario accuracy A_{US} at node 36. In some embodiments, input samples are sorted into sets based on selected user scenarios. Collective accuracy is then calculated for each set. In some embodiments, accuracy for each set is calculated using Equations 1 through 3. If the scenario involves multiple applications, the original ink provided by the user is compared against the ultimate recognition result provided by the destination application. In other embodiments, other measures of accuracy are used. For example, in a user scenario involving creation and recognition of ink in one application and transfer of the recognition result to another application, accuracy may be measured based on how the recognition result is handled by the subsequent application (e.g., 1 if the recognition result is accepted by the subsequent application, 0 otherwise). In some cases, accuracy is based upon how deep into a list of suggested recognition results a user must go to select the desired result. For example, a recognition result which is at the

top of a list of recognizer-suggested results would be given a value of 1.0, the second result on the list given a lower score, the third result on the list given an even lower score, etc.

[56] The collective accuracy for each set (however calculated) is then provided as an input (U1, U2, ... Un) to the user scenario accuracy calculation. Each of these inputs is weighted by a respective weighting factor υ_1 through υ_n. Weighting factors υ_1 through υ_n total 1.0. Each weighting factor υ represents the relative importance assigned to a particular user scenario, and may be determined in a variety of manners (e.g., usability studies, user questionnaires or other user research, product planning, etc.). As but one example, and assuming 3 possible user scenarios L, M and N, it might be estimated (or empirically determined) that 80% of operations as a whole will be in scenario L, 20% in scenario M and 5% in scenario N, giving respective values for υ_1, υ_2 and υ_3 of .80, .15 and .05. Of course, the weights υ need not be based on percentage of operations performed within a particular scenario. As but another example, weights υ could be based (in whole or part) upon the critical nature of a particular scenario (e.g., correctly recognizing a prescription in an application designed for general purpose use in a hospital).

[57] After multiplying by a weighting factor υ, the collective accuracy U for each user scenario is summed according to Equation 14.

(Equation 14) $$A_{US} = \sum_{i=1}^{n} \upsilon_i U_i$$

The counter n in Equation 14 corresponds to the number of user scenarios for which an accuracy was calculated and input into node 36.

[58] In some embodiments, data used to calculate the user scenario accuracy A_{US} is collected separately from user input sample data used for accuracy evaluation at other nodes. In at least one embodiment, users are asked to provide sample input with little

supervision. After data collection, the user scenarios are determined for the samples (by, e.g., software that recorded each software application opened and operation performed during an input).

[59] FIG. 15 illustrates calculation of overall accuracy A_T at node 32. In particular, node 32 sums the weighted outputs of nodes 34 (word accuracy A_W) and 36 (user scenario A_{US}), as shown in Equation 15.

(Equation 15) $$A_T = \Sigma [\gamma_1 * A_W + \gamma_2 * A_{US} + ... + \gamma_n * B( )]$$

The accuracies A_W and A_{US} are weighted by respective weight factors γ_1 and γ_2. The values of γ_1 and γ_2 total 1.0. Each weight γ corresponds to an importance assigned to a particular accuracy subcomponent of overall accuracy A_T. As with other weight factors, these values may be calculated in various manners. As one example, an accuracy model for recognizer intended for use in a software product that will have limited interaction with other software programs could assign a heavier weight of γ_1 relative to γ_2 than an accuracy model for a software product which must be integrated with numerous other diverse applications. Weights γ_1 and γ_2 could also be developed based on user research and/or on other sources such as previously discussed. Equation 15 includes additional terms beyond those supplied by nodes 34 through 36. In particular, Equation 15 continues with the expansion "+ ... + γ_n*B( )". Other embodiments may have additional subcomponents for overall accuracy A_T (shown generically as B( )) weighted by a separate weighting factor (γ_n). In the embodiment of FIG. 3, there are no other sub-components of overall accuracy A_T (i.e., γ_3 through γ_n equal 0). The values γ_1 for through γ_n total 1.0.

[60] In some embodiments, a transforming function is applied to some or all of the nodes of an accuracy model. This may be done in order to simplify a particular accuracy model or its implementation, and/or to prevent a particular accuracy node from being hidden. For example, a particular node may have a relatively small weight. If the accuracy at that node is very poor, the effect on overall accuracy A_T might not be

noticed during evaluation of a recognizer. Although the node may have a small relative weight, complete failure of the recognizer under the circumstances corresponding to the node may not be acceptable. Accordingly, a transforming function such as Equation 16 is applied to the output of the node.

(Equation 16) $$f(x)=\begin{cases} x, & x \geq x_{Threshold} \\ -M, & x < x_{Threshold} \end{cases}$$

A transforming function such as Equation 16 directly reflects the node sum until the sum decreases below a minimum value $x_{Threshold}$ (which value can be different for every node). If the sum falls below the minimum value, the $-M$ value (a very large number) is propagated through the model to A_T. In other words, failure of the node causes the entire recognizer to fail. As an example, it may be decided that a minimum accepted accuracy for e-mail recognition is 0.2. If the sum at the node corresponding to e-mail recognition is 0.1, the recognizer will have a very poor A_T, even if all other nodes have excellent recognition results. FIG. 16 illustrates application of a transform function to a node. In the example of FIG. 16, the transforming function is only shown applied to the demographic accuracy subcomponent (node 44) of accuracy model 30. However, a transforming function could be applied in a like manner to some or all of the nodes of an accuracy model. Indeed, transform functions need not be applied to all nodes. In some embodiments, nodes not having a minimum accuracy score simply assign a negative value to $x_{Threshold}$ or use a linear transforming function.

[61] In certain embodiments, each node of an accuracy model also has a confidence score. Those confidence scores can be applied to the nodes to determine an overall confidence score for the entire model. For example, referring to FIG. 4, there may be a 50% confidence level that the demographic group corresponding to input D1 is really a sufficiently large part of a user population to warrant a particular value for weight μ_1. By way of illustration, assume D1 corresponds to right-handed females (per the previous example), that right-handed persons constitute 85% of the population, and that the value assigned to μ_1 is .64 (.85*.75). However, there may be

insufficient data to have complete confidence in the conclusion that right-handed females are really 75% of the intended user population, resulting in only a 50% confidence level in this conclusion. Other inputs and nodes of the model may also have confidence levels less than 100%. For example, input scope 2 may correspond to e-mail addresses. Based on an assumption that users write e-mail addresses 10% of the time they spend writing, ξ_2 is given a value of .10. However, there may only be a 70% confidence level that users only spend 10% of their total writing time writing e-mail addresses.

[62] To determine an overall confidence score for the entire model, each collective accuracy input (D1-Dn in FIG. 4, S1-Sn in FIG. 5, α1-αn in FIG. 6, C1-Cn in FIG. 7, F1-Fn in FIG. 8, α1-αn in FIG. 11, SP1-SPn in FIG. 12, and US1-USn in FIG. 14) is set to 1. In other words, it is assumed that the recognition engine is 100% accurate. Each weight is then multiplied by its corresponding confidence level. For example, μ_1 is multiplied by .50 to yield .32 (.64*.50), ξ_2 is multiplied by .70 to yield .07 (.10*.70), etc. The resulting score at node 32 is an overall confidence level for the model.

[63] The overall confidence score is useful in various ways. As one example, a low overall confidence score and a high A_T indicate that additional data is needed for model validation. As another example, there may be relatively high confidence levels in most of the weighting factors in a model. However, because of insufficient data in one or more selected areas (e.g., in a particular demographic group of interest, with regard to a particular input scope, etc.), some of the weighting factors may have relatively low confidence values. By computing an overall confidence score for the model, it is thus possible to determine whether a relatively low confidence in certain weights causes the entire model to have a low confidence score. If not, the lack of data in those few areas is unlikely to affect the reliability of A_T for the recognizer. If so, the lack of data in those few areas indicates that an A_T computed using the accuracy model is suspect.

[64] As shown in FIGS. 17 and 18, an accuracy model according to the invention is inherently extensible in at least two ways. As shown in FIG. 17, additional nodes may be added to a particular summing node. Nodes 44, 46, 48, 50 and 52 in FIG. 17 correspond to like-numbered nodes in FIG. 3. However, it is assumed in FIG. 17 that ink color is found to have a non-trivial effect on accuracy for a handwriting recognizer. An additional node 54 is thus added to each input scope to include a color accuracy Co( ) component of each input scope, and a Co(IS) component is included at each of nodes 38_1 through 38_n. Other nodes could be added elsewhere in the model in a similar manner.

[65] As shown in FIG. 18, one or more nodes may further be broken out into additional levels of subcomponents. For example, instead of a single node for demographic accuracy D as shown in FIG. 3, one or more of the inputs D1-Dn can represent a weighted sum. By way of illustration, assume that D1 corresponds to a gender demographic, D2 corresponds to an age demographic, and Dn corresponds to a native language demographic. D1 is further broken down by male (G1) and female (G2), which are respectively weighted μ_1 and μ_2. D2 is further broken down by age groups (Age1 for Ages 10-20, Age2 for Ages 20-35, Age3 for Ages 35-50), which are respectively weighted by μ_1, μ_2 and μ_3. Dn is further broken down by specific nationalities (N) weighted by μ_1 through μ_n. Other nodes may be likewise subdivided.

[66] As previously described, an accuracy model according to the invention is also usable in connection with other recognition engines. FIG. 19 illustrates a sample database similar to that of FIG. 2, but gives examples of variable values for speech input. Similar to table 20 of FIG. 2, table 20' of FIG. 19 includes a separate row for each input sample, with each column corresponding to a different variable or collection of variables. The first column (Sample ID) corresponds to an identifier for a sample. The second column (Information) corresponds to the actual words the user intended to convey with his or her speech input. The next column (Demographic) contains demographic data about the provider of the input. Although similar in concept to the

demographic variable of table 20, the demographic groupings used for handwriting accuracy analysis would not necessarily be used for speech recognition accuracy analysis. For example, it might be determined that certain demographic groups speak less audibly than others.

[67] The next column (Input Scope), similar to the Input Scope column of table 20, also provides the context for the information the user is conveying. As with the demographic variable, the groupings for input scope used in handwriting accuracy analysis would not necessarily be used for speech recognition accuracy analysis. The Space column corresponds to a temporal interval between portions of a speech input (e.g., milliseconds between words, total duration of speech, etc.). The Scale column corresponds to an amount by which the volume of a speech input must be raised or lowered before processing by a recognizer. The user scenario column is similar to the user scenario column of table 20, but would not necessarily contain the same user scenarios. The content column, similar to the content column of table 20, represents various other types of data for a speech sample (e.g., program in which sample provided, particular aspect suggesting user is attempting to more clearly pronounce words, etc.). The format column corresponds to the format in which the sound and/or text component of a speech sample is stored. As in table 20 (FIG. 2), the columns of table 20' are merely examples of data regarding an input sample that may be collected and used for accuracy analysis in an accuracy model according to the invention. In some embodiments, some of the variables in table 20' are not used. In other embodiments, different variables are used.

[68] FIG. 20 is an accuracy model 30', according to at least one embodiment of the invention, for analyzing speech recognizer accuracy using the variables of table 20'. Model 30' operates similar to model 30 of FIGS. 3-15. In particular, collective accuracies are computed (using, e.g., Equations 1 through 3) for sample sets sorted by variable values. The collective accuracies are then weighted and summed. In the embodiment of model 30', demographic accuracy D'(IS1), scaling accuracy S'(IS1),

content accuracy C'(IS1) and format accuracy F'(IS1) for input scope 1 are calculated using adapted (e.g., substituting μ_i' for μ_i, etc.) versions of Equations 4, 5, 7 and 8, accuracy for input scope 1 is calculated using an adapted version of Equation 9, etc. Weighting factors (μ', π', ξ', etc.) likewise represent importance assigned to a particular accuracy sub-component, input scope, user scenario, etc.

[69] A transform function (FIG. 16) may be applied to one or more nodes of model 30'. Similarly, confidence scores for the weights of model 30' may also be used to determine an overall confidence score for model 30'.

General Purpose Computing Environment

[70] FIG. 21 illustrates a schematic block diagram of an exemplary conventional general-purpose digital computing environment 1000 that can be used to implement various aspects of the invention. The invention may also be implemented in other versions of computer 1000, such as a Tablet PC. The invention may also be implemented in connection with a multiprocessor system, a network PC, a minicomputer, a mainframe computer, hand-held devices, and the like.

[71] Computer 1000 includes a processing unit 1010, a system memory 1020, and a system bus 1030 that couples various system components including the system memory to the processing unit 1010. The system bus 1030 may be any of various types of bus structures using any of a variety of bus architectures. The system memory 1020 includes read only memory (ROM) 1040 and random access memory (RAM) 1050.

[72] A basic input/output system 1060 (BIOS), which is stored in the ROM 1040, contains the basic routines that help to transfer information between elements within the computer 1000, such as during start-up. The computer 1000 also includes a hard disk drive 1070 for reading from and writing to a hard disk (not shown), a magnetic disk drive 1080 for reading from or writing to a removable magnetic disk 1090, and an optical disk drive 1091 for reading from or writing to a removable optical disk 1082

such as a CD ROM, DVD or other optical media. The hard disk drive 1070, magnetic disk drive 1080, and optical disk drive 1091 are connected to the system bus 1030 by a hard disk drive interface 1092, a magnetic disk drive interface 1093, and an optical disk drive interface 1094, respectively. The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for computer 1000. It will be appreciated by those skilled in the art that other types of computer readable media may also be used.

[73] A number of program modules can be stored on the hard disk drive 1070, magnetic disk 1090, optical disk 1082, ROM 1040 or RAM 1050, including an operating system 1095, one or more application programs 1096, other program modules 1097, and program data 1098. A user can enter commands and information into the computer 1000 through input devices such as a keyboard 1001 and/or a pointing device 1002. These and other input devices are often connected to the processing unit 1010 through a serial port interface 1006 that is coupled to the system bus, but may be connected by other interfaces, such as a parallel port, game port, a universal serial bus (USB) or a BLUETOOTH interface. A monitor 1007 or other type of display device is also connected to the system bus 1030 via an interface, such as a video adapter 1008.

[74] In one embodiment, a pen digitizer 1065 and accompanying pen or stylus 1066 are provided in order to digitally capture freehand input. Although a direct connection between the pen digitizer 1065 and the processing unit 1010 is shown, in practice, the pen digitizer 1065 may be coupled to the processing unit 1010 via a serial port, parallel port or other interface and the system bus 1030, as known in the art. Furthermore, although the digitizer 1065 is shown apart from the monitor 1007, the usable input area of the digitizer 1065 is often co-extensive with the display area of the monitor 1007. Further still, the digitizer 1065 may be integrated in the monitor

1007, or may exist as a separate device overlaying or otherwise appended to the monitor 1007.

Conclusion

[75] Although specific examples of carrying out the invention have been described, those skilled in the art will appreciate that there are numerous variations and permutations of the above described systems and techniques that fall within the spirit and scope of the invention as set forth in the appended claims. As but one example, one or more nodes of the model 30 or model 30' (or of another accuracy model) are rearranged in some embodiments. By way of illustration, a spacing accuracy SP may be calculated for specific input scopes, and/or a demographic accuracy D may be calculated across all input scopes. An accuracy model according to the invention could also be used to measure the improvement in overall accuracy after a particular recognizer has been "personalized," i.e. modified for use by a specific user. If one or more words or phrases have a special importance (such that they must be correctly recognized for a recognizer to be considered accurate), those words or phrases can be assigned separate nodes and transforming functions applied to those nodes. These and other modifications are within the scope of the invention as defined by the attached claims.